AMENDMENTS TO THE CLAIMS

Please amend Claims 1 and 16 as follows:

1   1.   (Currently Amended) A process for routing packets through a load balancing array

2   of servers across a network in a computer environment, comprising the steps of:

3   requesting, by a scheduler, assign[[ing]]ment of a virtual IP address to [[a]] said

4   scheduler, said scheduler that is designated as active scheduler for a load balancing array;

5   wherein all incoming packets from requesting clients destined for the load balancing

6   array are routed through said scheduler via the virtual IP address;

7   wherein said scheduler routes and load balances a request packet from a requesting

8   client to a load balancing server;

9   wherein said load balancing server routes and load balances said request packet to a

10   back end Web server;

11   wherein said back end Web server's response packet to said request packet is sent to

12   said load balancing server; and

13   wherein said load balancing server sends said response packet directly to said client.

1   2.   (Previously Presented) The process of Claim 1, wherein said scheduler is a load

2   balancing server and routes and load balances client requests to itself.

1   3.   (Previously Presented) The process of Claim 1, further comprising the steps of:

2   detecting the failure of said scheduler; and

3   electing a load balancing server among a plurality of load balancing servers as the a

4   new scheduler.

1   4.   (Original) The process of Claim 1, wherein said scheduler detects the failure of

2   other load balancing servers; and wherein said scheduler stops routing packets to any failed

3   load balancing servers.

1   5.   (Original) The process of Claim 1, wherein said load balancing server schedules

2   sessions to back end Web servers based on a cookie or session ID.

1   6.   (Original) The process of Claim 1, wherein said load balancing server uses cookie

2   injection to map a client to a specific back end Web server.

1   7.   (Original) The process of Claim 1, wherein said load balancing server decrypts said

2   request packet if it is an SSL session before routing and load balancing said request packet

3   to a back end Web server.

1   8.   (Original) The process of Claim 7, wherein said load balancing server encrypts said

2   response packet if it is an SSL session before sending said response packet directly to said

3   client.

1   9.   (Original) The process of Claim 1, wherein said load balancing server establishes a

2   connection with said client and said client keeps said connection alive with said load

3   balancing server.

1   10.   (Original)   The process of Claim 9, wherein said load balancing server performs

2   URL based scheduling of request packets.

1   11.   (Original)   The process of Claim 9, wherein said load balancing server performs

2   hash scheduling of request packets.

1   12.   (Previously Presented)   The process of Claim 1, wherein said load balancing server

2   maintains persistent connections in its paths when required; and wherein said load balancing

3   server uses hash group based persistence to maintain its persistence tables.

1   13.   (Original)   The process of Claim 1, wherein said load balancing server detects if a

2   back end Web server fails; and wherein said load balancing server stops routing request

3   packets to failed back end Web servers.

1   14.   (Previously Presented)   The process of Claim 1, further comprising the step of:

2   providing a content delivery network; and

3   wherein said load balancing server modifies select URLs in an HTML page in said

4   response packet to serve the select URLs from said content delivery network.

1   15.   (Original)   The process of Claim 14, wherein HTML pages that have modified URLs

2   are cached to improve performance.

1   16.   (Currently Amended)   An apparatus for routing packets through a load balancing

2   array of servers across a network in a computer environment, comprising:

3   <u>a module for a requesting, by a scheduler,</u> assign[[ing]]<u>ment of</u> a virtual IP address

4   to [[a]] <u>said</u> scheduler<u>, said scheduler</u> ~~that~~ is designated as active scheduler for a load

5   balancing array;

6   wherein all incoming packets from requesting clients destined for the load balancing

7   array are routed through said scheduler via the virtual IP address;

8   wherein said scheduler routes and load balances a request packet from a requesting

9   client to a load balancing server;

10   wherein said load balancing server routes and load balances said request packet to a

11   back end Web server;

12   wherein said back end Web server's response packet to said request packet is sent to

13   said load balancing server; and

14   wherein said load balancing server sends said response packet directly to said client.

1   17.   (Previously Presented) The apparatus of Claim 16, wherein said scheduler is a load

2   balancing server and routes and load balances client requests to itself.

1   18.   (Previously Presented) The apparatus of Claim 16, further comprising:

2   a module for detecting the failure of said scheduler; and

3   a module for electing a load balancing server among a plurality of load balancing

4   servers as ~~the~~ <u>a</u> new scheduler.

1   19.   (Original) The apparatus of Claim 16, wherein said scheduler detects the failure of

2   other load balancing servers; and wherein said scheduler stops routing packets to any failed

3   load balancing servers.

1   20.   (Original) The apparatus of Claim 16, wherein said load balancing server schedules

2   sessions to back end Web servers based on a cookie or session ID.

1   21.   (Original) The apparatus of Claim 16, wherein said load balancing server uses

2   cookie injection to map a client to a specific back end Web server.

1   22.   (Original) The apparatus of Claim 16, wherein said load balancing server decrypts

2   said request packet if it is an SSL session before routing and load balancing said request

3   packet to a back end Web server.

1   23.   (Original) The apparatus of Claim 22, wherein said load balancing server encrypts

2   said response packet if it is an SSL session before sending said response packet directly to

3   said client.

1   24.   (Original) The apparatus of Claim 16, wherein said load balancing server establishes

2   a connection with said client and said client keeps said connection alive with said load

3   balancing server.

1   25.   (Original) The apparatus of Claim 24, wherein said load balancing server performs

2   URL based scheduling of request packets.

1   26.   (Original) The apparatus of Claim 24, wherein said load balancing server performs

2   hash scheduling of request packets.

1 27. (Previously Presented) The apparatus of Claim 16, wherein said load balancing

2 server maintains persistent connections in its paths when required; and wherein said load

3 balancing server uses hash group based persistence to maintain its persistence tables.

1 28. (Original) The apparatus of Claim 16, wherein said load balancing server detects if a

2 back end Web server fails; and wherein said load balancing server stops routing request

3 packets to failed back end Web servers.

1 29. (Previously Presented) The apparatus of Claim 16, further comprising:

2 a content delivery network; and

3 wherein said load balancing server modifies select URLs in an HTML page in said

4 response packet to serve the select URLs from said content delivery network.

1 30. (Original) The apparatus of Claim 29, wherein HTML pages that have modified

2 URLs are cached to improve performance.